

日 本 国 特 許 庁
JAPAN PATENT OFFICE

12.11.2004

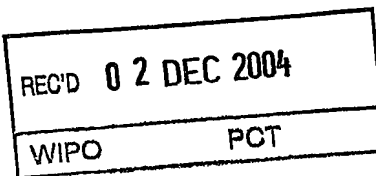
別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 1 0 月 2 9 日
Date of Application:

出 願 番 号 特 願 2 0 0 3 - 3 6 9 1 7 6
Application Number:
[ST. 10/C]: [J P 2 0 0 3 - 3 6 9 1 7 6]

出 願 人 日 本 電 気 株 式 有 限 公 司
Applicant(s):

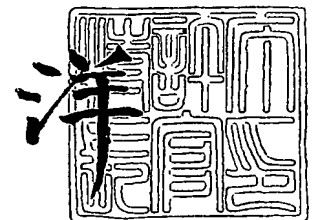


PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

2 0 0 4 年 8 月 2 3 日

特許庁長官
Commissioner,
Japan Patent Office

小 川



【書類名】 特許願
【整理番号】 34403309
【あて先】 特許庁長官殿
【国際特許分類】 H04N 7/50
【発明者】
 【住所又は居所】 東京都港区 5 丁目 7 番 1 号 日本電気株式会社内
 【氏名】 仙田 裕三
【特許出願人】
 【識別番号】 000004237
 【氏名又は名称】 日本電気株式会社
【代理人】
 【識別番号】 100079005
 【弁理士】
 【氏名又は名称】 宇高 克己
【手数料の表示】
 【予納台帳番号】 009265
 【納付金額】 21,000円
【提出物件の目録】
 【物件名】 特許請求の範囲 1
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1
 【包括委任状番号】 9715827

【書類名】 特許請求の範囲**【請求項 1】**

二値算術符号の入力に応じて、二値算術符号を復号し、二値シンボルを得る算術符号復号手段と、

前記復号された二値シンボルを蓄積するバッファと、

前記バッファから二値シンボルを取り出す際、自身の出力に応じて二値シンボルを取り出し、多値シンボルに変換して出力する逆二値化変換手段と
を有することを特徴とする二値化算術符号の復号器。

【請求項 2】

多値シンボルの入力に応じて、多値シンボルを二値シンボルに変換する二値化変換手段と、

前記二値シンボルを蓄積するバッファと、

前記バッファから二値シンボルを取り出す際、自身の出力に応じて二値シンボルを取り出して二値算術符号を生成する算術符号化手段と
を有することを特徴とする二値化算術符号の符号化器。

【請求項 3】

算術符号の入力に応じて、算術符号を復号して多値シンボルを得る算術符号復号手段と、

前記多値シンボルを蓄積するバッファと、

前記バッファから多値シンボルを取り出す際、自身の出力に応じて多値シンボルを取り出して出力シンボルに変換して出力する逆変換手段と
を有することを特徴とする算術符号の復号器。

【請求項 4】

入力シンボルの入力に応じて、入力シンボルを多値シンボルに変換する変換手段と、

前記多値シンボルを蓄積するバッファと、

前記バッファから多値シンボルを取り出す際、自身の出力に応じて多値シンボルを取り出して算術符号を生成する算術符号化手段と
を有することを特徴とする算術符号の符号化器。

【請求項 5】

請求項 2 記載の二値化算術符号の符号化器において、前記算術符号化手段が取り出した二値シンボル数と生成した符号のビット数とから二値シンボル数と符号ビット数との関係を推定し、前記バッファの蓄積量から算術符号化後に生成される符号ビット数を推定するビット数推定手段を持つことを特徴とする二値化算術符号の符号化器。

【請求項 6】

請求項 4 記載の算術符号の符号化器において、前記算術符号化手段が取り出した多値シンボル数と生成した符号のビット数とから多値シンボル数と符号ビット数との関係を推定し、前記バッファの蓄積量から算術符号化後に生成される符号ビット数を推定するビット数推定手段を持つことを特徴とする算術符号の符号化器。

【請求項 7】

復号された二値シンボルを蓄積するバッファを有する復号器における復号方法であって、
二値算術符号の入力に応じて、二値算術符号を復号し、二値シンボルを得る算術符号復号ステップと、

前記バッファから二値シンボルを取り出す際、自身の出力に応じて二値シンボルを取り出し、多値シンボルに変換して出力する逆二値化変換ステップと
を有することを特徴とする二値化算術符号の復号方法。

【請求項 8】

変換された二値シンボルを蓄積するバッファを有する符号化器における符号化方法であって、

多値シンボルの入力に応じて、多値シンボルを二値シンボルに変換する二値化変換ステップと、

前記バッファから二値シンボルを取り出す際、自身の出力に応じて二値シンボルを取り

出して二値算術符号を生成する算術符号化ステップと
を有することを特徴とする二値化算術符号の符号化方法。

【請求項 9】

復号された多値シンボルを蓄積するバッファを有する復号器における複合方法であって、
算術符号の入力に応じて、算術符号を復号して多値シンボルを得る算術符号復号ステップと、

前記バッファから多値シンボルを取り出す際、自身の出力に応じて多値シンボルを取り出して出力シンボルに変換して出力する逆変換ステップと
を有することを特徴とする算術符号の復号方法。

【請求項 10】

変換された多値シンボルを蓄積するバッファを有する符号化器における符号化方法であって、

入力シンボルの入力に応じて、入力シンボルを多値シンボルに変換する変換ステップと

、
前記バッファから多値シンボルを取り出す際、自身の出力に応じて多値シンボルを取り出して算術符号を生成する算術符号化ステップと
を有することを特徴とする算術符号の符号化方法。

【請求項 11】

請求項 8 記載の二値化算術符号の符号化方法において、

取り出した二値シンボル数と生成した符号のビット数とから二値シンボル数と符号ビット数との関係を推定し、前記バッファの蓄積量から算術符号化後に生成される符号ビット数を推定するビット数推定方法を有することを特徴とする二値化算術符号の符号化方法。

【請求項 12】

請求項 10 記載の算術符号の符号化方法において、取り出した多値シンボル数と生成した符号のビット数とから多値シンボル数と符号ビット数との関係を推定し、前記バッファの蓄積量から算術符号化後に生成される符号ビット数を推定するビット数推定方法を有することを特徴とする算術符号の符号化方法。

【請求項 13】

復号された二値シンボルを蓄積するバッファを有するコンピュータのプログラムであって

、
前記プログラムは前記コンピュータを、
二値算術符号の入力に応じて復号して二値シンボルを得る算術符号復号手段と、
前記バッファから二値シンボルを取り出す際、自身の出力に応じて二値シンボルを取り出し、多値シンボルに変換して出力する逆二値化変換手段と
して機能させることを特徴とするプログラム。

【請求項 14】

変換された二値シンボル列を蓄積するバッファを有するコンピュータのプログラムであって、

前記プログラムは前記コンピュータを、
多値シンボルの入力に応じて二値シンボル列に変換する二値化変換手段と、
前記バッファから二値シンボルを取り出す際、自身の出力に応じて二値シンボルを取り出し、二値算術符号を生成する算術符号化手段と
して機能させることを特徴とするプログラム。

【請求項 15】

復号された多値シンボルを蓄積するバッファを有するコンピュータのプログラムであって

、
前記プログラムは前記コンピュータを、
算術符号の入力に応じて復号し、多値シンボルを得る算術符号復号手段と、
前記バッファから多値シンボルを取り出す際、自身の出力に応じて多値シンボルを取り出し、出力シンボルに変換して出力する逆変換手段と

して機能させることを特徴とするプログラム。

【請求項 16】

変換された多値シンボル列を蓄積するバッファ有するコンピュータのプログラムであって

、
前記プログラムは前記コンピュータを、
入力シンボルの入力に応じて多値シンボル列に変換する変換手段と、
前記バッファから多値シンボル列を取り出す際、自身の出力に応じて多値シンボルを取り出し、算術符号を生成する算術符号化手段として機能させることを特徴とするプログラム。

【請求項 17】

前記プログラムは、前記算術符号化手段が取り出した二値シンボル数と生成した符号のビット数とから二値シンボル数と符号ビット数との関係を推定し、前記バッファの蓄積量から算術符号化後に生成される符号ビット数を推定するビット数推定手段として更に機能させることを特徴とする請求項 14 に記載のプログラム。

【請求項 18】

前記プログラムは、前記算術符号化手段が取り出した多値シンボル数と生成した符号のビット数とから多値シンボル数と符号ビット数との関係を推定し、前記バッファの蓄積量から算術符号化後に生成される符号ビット数を推定するビット数推定手段として更に機能させることを特徴とする請求項 16 に記載のプログラム。

【書類名】明細書

【発明の名称】算術符号の復号器、符号化器、復号方法、及び符号化方法

【技術分野】

【0001】

本発明は、算術符号の復号および符号化に関する。特に、多値シンボルを二値化した二値シンボル列の算術符号の復号および符号化の実装に関する。

【背景技術】

【0002】

二値化算術符号化は圧縮符号化技術の一つである。二値化算術符号化では、一つの多値シンボルを二値化して二値シンボル列を生成し、この二値シンボル列を算術符号化することで、最終的な二値化算術符号を得る。算術符号はハフマン符号などと比較すると処理コストが高く、これまでファイル圧縮や静止画圧縮などの実時間性が要求されないアプリケーションで利用されてきた。ところが近年のLSIの高速化に伴い、映像の符号化にも採用されるようになった。その一つにInternational Telecommunication Union Telecommunication Standardization Sector (ITU-T)が制定している新しいビデオコーデックの国際標準規格H.264のMain Profileがある。

【0003】

H.264では二値化算術符号をContext-based Adaptive Binary Arithmetic Coding (CABAC: キャバック)と呼んでいる。CABACの詳細については、IEEE学会のInternational Conference on Image Processing (ICIP)においてD. Marpeらが、2002年会合でContext-based adaptive binary arithmetic coding in JVT/H.26Lという題で、2001年会合ではVideo compression using context-based adaptive arithmetic codingという題で報告している。

【0004】

CABACでは、まず符号化すべき多値シンボルを二値シンボル(Bin)の列に二値化変換(Binarization)し、各BinをBinごとに定められたコンテキスト(Context)に対する確率推定値に従って二値算術符号化する。二値化変換は数値を数式で規定されたフォーマットに合わせて多値をビットパターンに変換するが、これは簡易な可変長符号化(VLC)と考えることができる。コンテキストの選択では、元々の多値シンボルが何を表しているか、周辺ブロックのパラメータ、二値シンボル列の何番目か、という状況が用いられる。逆に復号では、現在復号しようとしている二値シンボルのコンテキストから確率推定値を求め、算術符号の復号を行う。二値シンボルが復元されれば、確率推定値の更新を行うとともに、次に復号すべき二値シンボルのコンテキストを選択する。

【0005】

理想的な算術符号化はデータをエントロピーの限界まで圧縮できるので、理論的には1ビットで無限のBinを表すことができる。ただしこのままでは実装が困難なので、CABACでは簡略化した算術符号化を採用するとともに、1ビット当たりの平均Bin数に上限を設けている。簡略化では乗算をテーブル引きで代用しており、一つのBinのデコードに必要な演算を、テーブル引き、比較、減算に抑えている。

【0006】

H.264 CABACのような二値化算術符号化では、算術符号の復号および符号化の処理コストが高い。

【0007】

図2にH.264復号器の全体構成を示す。

【0008】

H.264復号器はストリームを受信して溜めておくCPBバッファと、各フレームをフレーム間隔毎に復号する瞬時復号器とから構成される。瞬時復号器は、CABAC復号器とブロック復号器とから構成される。ブロック復号器は、逆量子化、逆離散整数変換、動き補償予測

、ループ内フィルタ処理などを行っており、画素数に比例した処理コストとなっている。

【0009】

これに対してCABAC復号器の処理コストはBin数に比例する。

【0010】

図3にCABAC復号器の詳細を示す。

【0011】

CABAC復号器は二値算術符号復号器、逆二値化変換器、コンテキストごとの確率推定値を保存するメモリと、これらを制御する制御器とから構成される。処理の単位はBinの復号であり、制御器はBinを復号するたびに、確率推定値を更新するとともに、H.264規格の文法に従って内部のステートを遷移させる。これらの処理は複数のBinをまとめて行うことができないため、Bin数が処理コストを決定する。

【0012】

ここで具体的にどれだけの処理コストになるのかを算出してみる。各フレームの圧縮率はフレームの符号化タイプ(フレーム内もしくはフレーム間)や予測的中度合い、画質によって異なるため、各フレームのビット数はフレームごとに変動する。つまり、CABAC復号器の処理コストはフレームごとに変動する。規格では1フレームの最大ビット数は $2048 * \text{MaxMBPS} * \text{DeltaTime} * \text{ChromaFormatFactor} / \text{MinCR}$ で与えられており、フレーム期間平均の最大ビットレートに換算すると $2048 * \text{MaxMBPS} * \text{ChromaFormatFactor} / \text{MinCR}$ となる。ここでMaxMBPSは一秒当たりの最大マクロブロック数、DeltaTimeはフレーム時間間隔、ChromaFormatFactorは輝度信号に対して色信号を加えた場合のサンプル数の比率、MinCRは最低圧縮率である。

【0013】

Annex A記載のLevel 4.1ではMaxMBPS=245760、ChromaFormatFactor=1.5、MinCR=2なので、最大ビットレートは377Mbpsとなる。Bin対ビットの圧縮率は1.33以下となるように規定されているので、最大Binレートに換算すると503Mbin/secとなる。最大ビットレートをフレーム期間平均で求めているので、ここでの最大Binレートはフレーム期間平均で処理すべきBinの数をフレーム期間で除した値となっている。復号器の性能がこの最大Binレートを達成できていないと、フレームを表示すべき時刻までに復号処理が終わらず、フレームの欠落といった大きな画質劣化を引き起こす。

【0014】

以上は復号器の実装について説明したが、逆の動作を行う符号化器も同様になる。

【0015】

図4にH.264符号化器の構成を示す。

【0016】

ブロック符号化器は入力された画素レートで、動き補償予測、離散整数変換、量子化、逆量子化、逆離散整数変換、ループ内フィルタ処理などを行う。その後、ブロック情報は二値化変換器でBin列に変換される。Bin列は算術符号化器で符号ビット列に変換され、出力バッファに送られる。出力バッファの蓄積量はブロック符号化器にフィードバックされ、ブロック符号化器内部の符号量制御に用いられる。

【0017】

二値化ではブロック情報の一つの要素、たとえば変換係数が複数のBinに変換される。そのため、Bin列の生成速度は瞬間的には画素レートの10倍以上になる。それ以後のBin列を扱う制御器やメモリ、算術符号化器はその速度で動く必要がある。フレーム期間での処理を考えると、H.264符号化器の最大BinレートはH.264復号器の場合と同じく503Mbin/secとなる。

【0018】

【非特許文献1】2001 IEEE International Conference on Image Processing, ISBN: 0-7803-6725-1, pages 558-561

【非特許文献2】2002 IEEE International Conference on Image Processing, ISBN: 0-7803-7623-4 IEEE Catalog No.:02CH37396, pages 2-513 - 2-536

【発明の開示】

【発明が解決しようとする課題】

【0019】

従来の技術では、高いビットレートでの実時間処理が依然として困難である。たとえば、H.264規格書の記載の通りに復号処理を行おうとすると、処理すべきBinレートは非現実的な値となる。H.264 Level 4.1の規定を満たす最大Binレートは503Mbin/secであり、一つのBinを2サイクルで処理できたとしても、1 GHz以上の周波数でCABAC復号器や算術符号化器を動作させなければならない。この値は現状のLSIで容易に、もしくは安価に実現可能な速度の数倍となっている。

【0020】

本発明の目的は、従来技術と比較して、低い最大Binレートで実時間での二値化算術符号の復号を行う復号器、および符号化を行う符号化器を提供することにある。

【課題を解決するための手段】

【0021】

上記本発明の目的を達成するための第一の発明は、二値算術符号の入力に応じて、二値算術符号を復号し、二値シンボルを得る算術符号復号手段と、前記復号された二値シンボルを蓄積するバッファと、前記バッファから二値シンボルを取り出す際、自身の出力に応じて二値シンボルを取り出し、多値シンボルに変換して出力する逆二値化変換手段とを有することを特徴とする。

【0022】

この構成において、前記算術符号復号手段と前記逆二値化変換手段とは別々に動作しており、通常異なる速度で処理を進める。

【0023】

上記本発明の目的を達成するための第二の発明は、多値シンボルの入力に応じて、多値シンボルを二値シンボルに変換する二値化変換手段と、前記二値シンボルを蓄積するバッファと、前記バッファから二値シンボルを取り出す際、自身の出力に応じて二値シンボルを取り出して二値算術符号を生成する算術符号化手段とを有することを特徴とする。

【0024】

この構成において、前記算術符号化手段と前記二値化変換手段とは別々に動作しており、通常異なる速度で処理を進める。前記算術符号化手段が達成すべき処理性能は出力符号レートの最大値で規定できる。一方、前記二値化変換手段では多値シンボルの単位で処理が可能のため、これが達成すべき処理性能は入力多値シンボルレートの最大値で規定できる。

【0025】

上記本発明の目的を達成するための第三の発明は、算術符号の入力に応じて、算術符号を復号して多値シンボルを得る算術符号復号手段と、前記多値シンボルを蓄積するバッファと、前記バッファから多値シンボルを取り出す際、自身の出力に応じて多値シンボルを取り出して出力シンボルに変換して出力する逆変換手段とを有することを特徴とする。

【0026】

上記本発明の目的を達成するための第四の発明は、入力シンボルの入力に応じて、入力シンボルを多値シンボルに変換する変換手段と、前記多値シンボルを蓄積するバッファと、前記バッファから多値シンボルを取り出す際、自身の出力に応じて多値シンボルを取り出して算術符号を生成する算術符号化手段とを有することを特徴とする。

【0027】

この構成において、前記算術符号復号手段と前記逆変換手段とは別々に動作しており、通常異なる速度で処理を進める。前記算術符号復号器手段が達成すべき処理性能は入力符号レートの最大値で規定できる。

【0028】

上記本発明の目的を達成するための第五の発明は上記第二の発明において、前記算術符号化手段が取り出した二値シンボル数と生成した符号のビット数とから二値シンボル数と

符号ビット数との関係を推定し、前記バッファの蓄積量から算術符号化後に生成される符号ビット数を推定するビット数推定手段を持つことを特徴とする。

【0029】

この構成において、前記算術符号化手段と前記変換手段とは別々に動作しており、通常異なる速度で処理を進める。

【0030】

上記本発明の目的を達成するための第六の発明は、上記第四の発明において、前記算術符号化手段が取り出した多値シンボル数と生成した符号のビット数とから多値シンボル数と符号ビット数との関係を推定し、前記バッファの蓄積量から算術符号化後に生成される符号ビット数を推定するビット数推定手段を持つことを特徴とする算術符号の符号化器。

【0031】

上記本発明の目的を達成するための第七の発明は、復号された二値シンボルを蓄積するバッファを有する復号器における復号方法であって、二値算術符号の入力に応じて、二値算術符号を復号し、二値シンボルを得る算術符号復号ステップと、前記バッファから二値シンボルを取り出す際、自身の出力に応じて二値シンボルを取り出し、多値シンボルに変換して出力する逆二値化変換ステップとを有することを特徴とする。

【0032】

上記本発明の目的を達成するための第八の発明は、変換された二値シンボルを蓄積するバッファを有する符号化器における符号化方法であって、多値シンボルの入力に応じて、多値シンボルを二値シンボルに変換する二値化変換ステップと、前記バッファから二値シンボルを取り出す際、自身の出力に応じて二値シンボルを取り出して二値算術符号を生成する算術符号化ステップとを有することを特徴とする。

【0033】

上記本発明の目的を達成するための第九の発明は、復号された多値シンボルを蓄積するバッファを有する復号器における複合方法であって、算術符号の入力に応じて、算術符号を復号して多値シンボルを得る算術符号復号ステップと、前記バッファから多値シンボルを取り出す際、自身の出力に応じて多値シンボルを取り出して出力シンボルに変換して出力する逆変換ステップとを有することを特徴とする。

【0034】

上記本発明の目的を達成するための第十の発明は、変換された多値シンボルを蓄積するバッファを有する符号化器における符号化方法であって、入力シンボルの入力に応じて、入力シンボルを多値シンボルに変換する変換ステップと、前記バッファから多値シンボルを取り出す際、自身の出力に応じて多値シンボルを取り出して算術符号を生成する算術符号化ステップとを有することを特徴とする。

【0035】

上記本発明の目的を達成するための第十一の発明は、上記第八の発明において、取り出した二値シンボル数と生成した符号のビット数とから二値シンボル数と符号ビット数との関係を推定し、前記バッファの蓄積量から算術符号化後に生成される符号ビット数を推定するビット数推定方法を有することを特徴とする二値化算術符号の符号化方法。

【0036】

上記本発明の目的を達成するための第十二の発明は、上記第十の発明において、取り出した多値シンボル数と生成した符号のビット数とから多値シンボル数と符号ビット数との関係を推定し、前記バッファの蓄積量から算術符号化後に生成される符号ビット数を推定するビット数推定方法を有することを特徴とする。

【0037】

上記本発明の目的を達成するための第十三の発明は、復号された二値シンボルを蓄積するバッファを有するコンピュータのプログラムであって、前記プログラムは前記コンピュータを、二値算術符号の入力に応じて復号して二値シンボルを得る算術符号復号手段と、前記バッファから二値シンボルを取り出す際、自身の出力に応じて二値シンボルを取り出し、多値シンボルに変換して出力する逆二値化変換手段として機能させることを特徴とする。

る。

【0038】

上記本発明の目的を達成するための第十四の発明は、変換された二値シンボル列を蓄積するバッファを有するコンピュータのプログラムであって、前記プログラムは前記コンピュータを、多値シンボルの入力に応じて二値シンボル列に変換する二値化変換手段と、前記バッファから二値シンボルを取り出す際、自身の出力に応じて二値シンボルを取り出し、二値算術符号を生成する算術符号化手段として機能させることを特徴とする。

【0039】

上記本発明の目的を達成するための第十五の発明は、復号された多値シンボルを蓄積するバッファを有するコンピュータのプログラムであって、前記プログラムは前記コンピュータを、算術符号の入力に応じて復号し、多値シンボルを得る算術符号復号手段と、前記バッファから多値シンボルを取り出す際、自身の出力に応じて多値シンボルを取り出し、出力シンボルに変換して出力する逆変換手段として機能させることを特徴とする。

【0040】

上記本発明の目的を達成するための第十六の発明は、変換された多値シンボル列を蓄積するバッファを有するコンピュータのプログラムであって、前記プログラムは前記コンピュータを、入力シンボルの入力に応じて多値シンボル列に変換する変換手段と、前記バッファから多値シンボル列を取り出す際、自身の出力に応じて多値シンボルを取り出し、算術符号を生成する算術符号化手段として機能させることを特徴とする。

【0041】

上記本発明の目的を達成するための第十七の発明は、上記第十四の発明において、前記プログラムは、前記算術符号化手段が取り出した二値シンボル数と生成した符号のビット数とから二値シンボル数と符号ビット数との関係を推定し、前記バッファの蓄積量から算術符号化後に生成される符号ビット数を推定するビット数推定手段として更に機能させることを特徴とする。

【0042】

上記本発明の目的を達成するための第十八の発明は、上記第十六の発明において、前記プログラムは、前記算術符号化手段が取り出した多値シンボル数と生成した符号のビット数とから多値シンボル数と符号ビット数との関係を推定し、前記バッファの蓄積量から算術符号化後に生成される符号ビット数を推定するビット数推定手段として更に機能させることを特徴とする。

【発明の効果】

【0043】

二値化算術符号の復号器および符号化器が達成すべき二値シンボル処理レートの最大値を大幅に下げることができる。本発明の算術符号復号手段が達成すべき処理性能は入力符号レートの最大値で規定でき、同様に本発明の算術符号の符号化手段が達成すべき処理性能は出力符号レートの最大値で規定できる。

【0044】

たとえばH.264のLevel 4.1に適用する場合、Max Video Bitrateが50Mbpsなので、最大Binレートは66.7Mbin/secとなる。この値は従来の技術の場合の7分の1以下となっており、実装が大幅に容易になることが分かる。

【0045】

従来技術ではCPBバッファにストリームを溜めていたが本発明では不要となる。その代わりにCPBバッファよりも若干大きなメモリ手段を必要としている。H.264の場合には算術符号の圧縮率が1.33以下となるように制限されているので、メモリ手段はCPBバッファの1.33倍の大きさがあればよい。

【0046】

本発明の符号化器は二値もしくは多値シンボルのバッファを持っているため、バッファ遅延分だけ遅れるが、ビット数推定手段が生成される実際の符号ビット数を得ることができるので、遅れの無い推定値をその代替として提供できる。また、ビデオ符号化器などの

符号量制御が必要な場合、遅延された生成ビット数を用いると制御が不安定になるが、本発明ではその推定値が使えるのでバッファ遅延の影響を抑圧できる。

【発明を実施するための最良の形態】

【0047】

本発明の算術符号の復号器は、二値算術符号の入力に応じて復号して二値シンボルを得る算術符号復号器と、二値シンボルを蓄積する中間バッファと、中間バッファから二値シンボル列を取り出して多値シンボルに変換して出力し、この多値シンボルの出力に応じて中間バッファから二値シンボル列を取り出す逆二値化変換手段とを備える。

【0048】

又、本発明における算術符号の符号化器は、多値シンボルの入力に応じて多値シンボルを二値シンボルに変換する二値化変換手段と、前記二値シンボルを蓄積するバッファと、前記バッファから二値シンボルを取り出す際、自身の出力に応じて二値シンボルを取り出して二値算術符号を生成する算術符号化手段とを備える。

【0049】

又、本発明における算術符号の復号器は、算術符号の入力に応じて算術符号を復号して多値シンボルを得る算術符号復号手段と、前記多値シンボルを蓄積するバッファと、前記バッファから多値シンボルを取り出す際、自身の出力に応じて多値シンボルを取り出して出力シンボルに変換して出力する逆変換手段とを備える。

【0050】

又、本発明における算術符号の符号化器は、入力シンボルの入力に応じて入力シンボルを多値シンボルに変換する変換手段と、前記多値シンボルを蓄積するバッファと、前記バッファから多値シンボルを取り出す際、自身の出力に応じて多値シンボルを取り出して算術符号を生成する算術符号化手段とを備える。

【実施例1】

【0051】

図1は、本発明の二値化算術符号の復号器を用いた映像復号器のブロック図である。

【0052】

算術符号復号器10は入力されたストリームの算術符号を復号して二値シンボル(Bin)を得て、制御器11、逆二値化変換機12に供給するとともに、中間バッファ14に格納する。復号に必要なコンテキストの確率推定値は制御器11から供給される。

【0053】

制御器11はストリームの文法に従い、現在復号すべき二値シンボルからコンテキストを選択し、メモリ13からその確率推定値を取得するとともに、確率推定値を算術符号復号器10に供給する。コンテキストの選択においては、必要であればメモリ13に格納されているブロック情報を用いる。制御器11は算術符号復号器10から二値シンボルを得ると、メモリ13に格納されている確率推定値を更新するとともに、二値シンボル列の構成情報を逆二値化変換器12に供給する。構成情報としては、多値シンボルの示しているパラメータ名、二値シンボル列のフォーマット情報、逆変換を行うタイミングなどがある。

【0054】

逆二値化変換器12は、算術符号復号器10から得られる二値シンボルと制御器11とから供給される構成情報から、必要に応じて二値シンボル列を多値シンボルに変換し、その結果として得られるブロック情報をメモリ13に格納する。ストリームに含まれるブロック情報には、量子化された変換係数、量子化パラメータ、有効ブロックパターン、予測モード、動きベクトルなどがあるが、メモリ13に格納すべきブロック情報は制御器11が参照するものである。

【0055】

中間バッファ14は、算術符号復号器10で得られた二値シンボルを格納し、後段の制御器15と逆二値化変換器16とに二値シンボルを供給する。また、制御器15に二値シンボルを供給する際、制御器15からの指示に基づいて、二値シンボルを供給する。

【0056】

制御器15はストリームの文法に従い、中間バッファ14から二値シンボル列を得て、構成情報を逆二値化変換器16に供給する。

【0057】

逆二値化変換器16は、中間バッファ14から二値シンボル列を得て、制御器15から供給される構成情報から、二値シンボル列を多値シンボルに変換し、その結果として得られるブロック情報をブロック復号器17に供給する。

【0058】

ブロック復号器17は、逆二値化変換器16から供給されるブロック情報に基づき、逆量子化、逆整数変換、動き補償予測、ループ内フィルタの処理を行うことで復号画像を得て、得られた復号画像を出力する。

【0059】

中間バッファ14よりも前段のブロックである算術符号復号器10、制御器11、逆二値化変換器12、及びメモリ13は、算術符号復号器10に入力されるストリームのビット、バイト又はバイト列等に合わせて処理を進めていく。これに対して、中間バッファ14よりも後段のブロックである制御器15、逆二値化変換器16、及びブロック復号器17は、復号画像の出力に合わせて処理を進める。このように算術符号復号器10と逆二値化変換器16とは別々に動作しており、通常異なる速度で処理を進めている。中間バッファ14は前後の処理速度の違いを吸収する。

【0060】

図5は本発明の二値化算術符号の符号化器を用いた映像符号化器のブロック図である。

【0061】

ブロック符号化器20は、ビット数推定器27から与えられる推定生成ビット数を参考にして、入力画像に対して動きベクトル探索、動き補償予測、離散整数変換、量子化、逆量子化、逆離散整数変換、ループ内フィルタの処理を行い、ブロック情報を生成する。ブロック情報はストリーム構成に必要な情報であり、量子化された変換係数、量子化パラメータ、有効ブロックパターン、予測モード、動きベクトルなどが含まれる。

【0062】

得られたブロック情報は二値化変換器21で二値シンボル列に変換され、その結果は中間バッファ22に格納される。

【0063】

中間バッファ22は、二値化変換器21で変換された二値シンボル列を格納し、算術符号化器25からの指示に基づいて、算術符号化器25に二値シンボルを供給する。また、中間バッファ22は、蓄積量をビット数推定器27に供給する。

【0064】

逆二値化変換器23は、中間バッファ22から得られる二値シンボル列と、制御器24から供給される構成情報とから、ブロック情報を復元し、メモリ26に格納する。ここで復元すべきブロック情報は制御器24が参照するものである。

【0065】

制御器24はストリームの文法に従い、現在符号化すべき二値シンボルからコンテキストを選択し、メモリ26からその確率推定値を取得するとともに、確率推定値を算術符号化器25に供給する。コンテキストの選択においては、必要であればメモリ26に格納されているブロック情報を用いる。中間バッファ22から二値シンボルを得ると、メモリ26に格納されている確率推定値を更新するとともに、二値シンボル列の構成情報を逆二値化変換器23に供給する。

【0066】

算術符号化器25は、中間バッファ22から得た二値シンボルと、制御器24から得た確率推定値とから二値算術符号化を行い、得られたストリームを出力する。また、算術符号化で読み出した二値シンボル数と生成した符号のビット数とをビット数推定器27に供給する。

【0067】

ビット数推定器 27 は、算術符号化器 25 から供給される二値シンボル数と符号ビット数とから、二値シンボル数と符号ビット数との関係を推定し、中間バッファ 22 から供給される蓄積量をビット数に換算して生成ビット数を求め、ブロック符号化器 20 に供給する。

【0068】

中間バッファ 22 よりも前段のブロックであるブロック符号化器 20、二値化変換器 21、及びビット数推定器 27 は、ブロック符号化器 20 に入力される画像のビット、バイト、バイト列に合わせて処理を進めていく。これに対して、中間バッファ 22 よりも後段のブロックである逆二値化変換器 23、制御器 24、算術符号化器 25、メモリ 26 はストリームの出力に合わせて処理を進める。このように二値化変換器 21 と算術符号化器 25 とは別々に動作しており、通常異なる速度で処理を進めている。中間バッファ 22 は前後の処理速度の違いを吸収する。

【0069】

図 6 は本発明の別の実施の形態を示すブロック図である。

【0070】

図 6 で本発明の二値算術符号の復号器を構成する場合、処理部 31 は算術符号の復号を行い、処理部 32 は逆二値化変換を行う。メモリ 33 は、処理部 31 と処理部 32 とからアクセス可能であり、処理部 31 の入力となる符号列、処理部 31 の出力かつ処理部 32 の入力となる二値シンボル列、処理部 32 の出力となる多値シンボル、処理で必要となる確率期待値やブロック情報などを保持する。

【0071】

尚、図 6 は論理的な構成を示すために処理部 31 と処理部 32 とを分けているが、オペレーティングシステムがマルチプロセス機能を提供している場合や、米 Intel 製 HyperThreading 対応 CPU のように単体プロセッサでマルチプロセス処理が可能な場合、物理的には一つとなる。また、メモリ 33 は物理的に単一メモリである必要はなく、処理部 31 からしかアクセスしない変数はバス接続でなく処理部 31 に直結していても構わない。

【0072】

次に図 7 を参照して、処理部 31 で算術符号の復号を行う場合の動作を説明する。

【0073】

符号化方式のシンタックスには、符号化モード、動きベクトル、コーデッドフラグ、係数が存在し、これらが二値算術符号化されていることを想定している。ここで、符号化モードからは動きベクトル情報の有無を知ることができ、コーデッドフラグは係数の有無を知ることができる。

【0074】

まず、ステップ A100 において初期化を行う。初期化では、コンテキストごとの確率推定値を初期値に設定する。ステップ A110 ではブロックの符号化モードを復号する。ステップ A111 では動きベクトル情報があるかどうかで分岐する。動きベクトル情報がある場合にはステップ A120 へ、動きベクトル情報がない場合にはステップ A130 へ進む。

【0075】

ステップ A120 では動きベクトル水平値を復号する。ステップ A121 では動きベクトル垂直値を復号する。ステップ A130 ではコーデッドフラグを復号する。ステップ A131 では係数があるかないかで分岐する。係数がある場合にはステップ A140 へ、係数がない場合にはステップ A150 へ進む。

【0076】

ステップ A140 では係数を復号する。ステップ A141 では係数が終了かどうかで分岐する。終了の場合にはステップ A150 へ、係数が続いている場合にはステップ A140 へ進む。

【0077】

ステップA150は符号列の終了かどうかで分岐する。終了でなければステップA110へ進む。

【0078】

これらのステップで行われる復号では、多値シンボルを復号するサブルーチンを呼び出している。このサブルーチンは図8の動作を行う。

【0079】

図8を参照して、サブルーチンを説明する。

【0080】

まずステップA10ではシンボル列バッファを空にする。ステップA11では現在のシンタックスに適合するコンテキストを選択する。必要であれば周辺ブロックの情報を用いる。ステップA12では現在のコンテキストの確率推定値を取得する。ステップA13では算術符号の復号を行う。符号の入力を待ち、入力されれば符号語の現在値と確率推定値とを比較して、大小関係から二値シンボルを得る。二値シンボルの0と1とに対して対称の動作をするのであれば、確率推定値をMPSの値とMPSの確率推定値とで表現(対称表現)することもできる。ここで、MPSとは発生確率の推定値の高いシンボルであり、MPSの確率推定値は0.5から1の値を取る。ステップA14では得られた二値シンボルをシンボル列バッファに蓄えとともに、メモリへ出力する。ステップA15では二値シンボルの値に応じて確率推定値を更新する。確率推定値が対称表現の場合、MPSの確率推定値が0.5未満となれば、MPSを反転する。ステップA16では、シンボル列バッファ内の二値シンボル列が、完結した二値シンボル列を構成しているかどうかで分岐する。完結した二値シンボル列を構成していればステップA17へ進み、そうでなければステップA11に戻って算術符号の復号を続ける。

【0081】

ステップA17では必要に応じて逆二値化変換を行う。必要となる条件としては、シンタックスに関わる要素やコンテキスト選択で参照される可能性がある要素がある。

【0082】

次に処理部32で逆二値化変換を行う場合の動作を説明する。全体の処理の流れは処理部31と同様に図7のフローで動作する。ただし、呼び出される復号サブルーチンが処理部31とは異なる。処理部32で用いる復号サブルーチンは逆二値変換であり、二値シンボル列から多値シンボルを復号するものとなる。処理部32は算術符号に関係しないので、ステップA100で確率推定値を初期値に設定する必要は無い。

【0083】

図6で本発明の二値算術符号の符号化器を構成する場合、処理部31は二値化変換を行い、処理部32は算術符号化を行う。メモリ33は、処理部31と処理部32とからアクセス可能であり、処理部31の入力となる多値シンボル、処理部31の出力かつ処理部32の入力となる二値シンボル列、処理部32の出力となる符号列、処理で必要となる確率期待値やブロック情報などを保持する。

【0084】

次に図9を参照して、処理部31で二値化変換を行う場合の動作を説明する。

【0085】

まず、ステップA200において初期化を行う。ステップA210ではブロックの符号化モードを符号化処理する。ステップA211では動きベクトル情報があるかどうかで分岐する。ある場合にはステップA220へ、ない場合にはステップA230へ進む。

【0086】

ステップA220では動きベクトル水平値を符号化処理する。ステップA221では動きベクトル垂直値を符号化処理する。ステップA230ではコードドフラグを符号化処理する。ステップA231では係数があるかないかで分岐する。係数がある場合にはステップA240へ、係数がない場合にはステップA250へ進む。

【0087】

ステップA240では係数を符号化処理する。ステップA241では係数終了かどうか

で分岐する。終了の場合にはステップA250へ、係数が続いている場合にはステップA240へ進む。

【0088】

ステップA250は符号化の終了かどうかで分岐する。符号化の終了でなければステップA210へ進む。これらのステップで行われる符号化処理は二値化変換であり、多値シンボルを二値シンボル列に変換し、出力する出力サブルーチンと呼び出している。出力サブルーチンでは、出力した二値シンボル数をカウントし、外部からの参照可能にしておく。

【0089】

次に処理部32で算術符号化を行う場合の動作を説明する。全体の処理の流れは処理部31と同様に図9のフローで動作する。ただし、ステップA200の初期化と呼び出される符号化処理サブルーチンが処理部31とは異なる。処理部32で用いる符号化処理サブルーチンでは、逆二値変換により二値シンボル列から多値シンボルを復号するとともに、二値シンボル列の算術符号化を行う。算術符号化を行うにあたり、ステップA200の初期化では確率期待値を初期値に設定しておく。

【0090】

図10を参照して、処理部32の符号化処理サブルーチンの動作を説明する。

【0091】

まずステップA20において、逆二値化変換を行い、多値シンボルを出力するとともに、対応する二値シンボル列をシンボル列バッファに格納する。ステップA21では現在のコンテキストの確率推定値を取得する。

【0092】

ステップA23では、シンボル列バッファの先頭から二値シンボルを一つ取り出し、算術符号化を行って出力する。算術符号化の回数と生成したビット数とをカウントし、外部からの参照可能にしておく。

【0093】

ステップA24では二値シンボルの値に応じて確率推定値を更新する。確率推定値が対称表現の場合、MPSの確率推定値が0.5未満となれば、MPSを反転する。ステップA25では、シンボル列バッファが空か否かで分岐する。シンボル列バッファが空であれば終了し、そうでなければステップA21に戻って算術符号化を続ける。

【0094】

処理部31では出力した二値シンボル数が、処理部32では算術符号化の回数と生成したビット数とが分かる。メモリ上に蓄積されている二値シンボル数は、出力した二値シンボル数から算術符号化の回数を減じることで求まる。また、二値シンボル数と符号ビット数との関係は、算術符号化の回数と生成したビット数とから求めることができる。これらの値から推定生成ビット数を算出することができる。推定生成ビット数は、処理部31の出力サブルーチン内で算出して外部から参照可能にしておいてもよいし、外部で元となる値から算出してもよい。

【0095】

次に、上述した二値算術符号の復号を行う復号器及び符号化器をコンピュータシステムにより実行する例を説明する。

【0096】

コンピュータシステムには、CPUが装備されており、CPUにはバッファ及びメモリが接続されている。

【0097】

メモリには、本発明における復号処理及び符号化処理を実行するためのプログラムが格納されている。このプログラムをCPUで実行することにより、本発明における復号処理及び符号化処理が実行される。

【0098】

尚、上述した実施の形態では二値算術符号を扱う場合について説明したが、本発明は二

値算術符号への適用に限られるものではない。四値の算術符号を用いる場合には、図および説明の二値を四値に読み替えるだけで、本発明の四値算術符号の復号器や符号化器を構成できる。また、二値と三値との算術符号が混在するような場合でも、コンテキストに合わせて二値算術符号の処理と三値算術符号の処理とを切り替えるように構成すればよい。

【0099】

以上、映像復号器と映像符号化器とを例にして、本発明の実施の形態を説明したが、本発明はこれらの適用に限られるものではない。ブロック復号器17を音声フレーム復号器、ブロック符号化器20を音声フレーム符号化器に置き換えれば、容易に音声復号器、音声符号化器への適用が可能である。他のメディアやデータの符号化器や復号器でも、二値化算術符号を用いるものであれば、映像や音声の場合と同様に、本発明の二値化算術符号の符号化器と復号器とを適用できる。

【産業上の利用可能性】

【0100】

本発明によれば、多値シンボルを二値化した二値シンボル列の算術符号の復号および符号化の実装に適用できる。

【図面の簡単な説明】

【0101】

【図1】図1は本発明の二値化算術符号の復号器を用いた映像復号器のブロック図である。

【図2】図2は国際標準規格ITU-T H.264復号器のブロック図である。

【図3】図3はH.264 CABAC復号器の内部のブロック図である。

【図4】図4はH.264符号化器のブロック図である。

【図5】図5は本発明の二値化算術符号の符号化器を用いた映像符号化器のブロック図である。

【図6】図6は本発明の二値化算術符号の復号器もしくは符号化器のブロック図である。

【図7】図7は本発明の二値化算術符号の復号処理を示すフローチャートである。

【図8】図8は本発明の復号サブルーチンのフローチャートである。

【図9】図9は本発明の二値化算術符号の符号化処理を示すフローチャートである。

【図10】図10は本発明の符号化処理サブルーチンのフローチャートである。

【符号の説明】

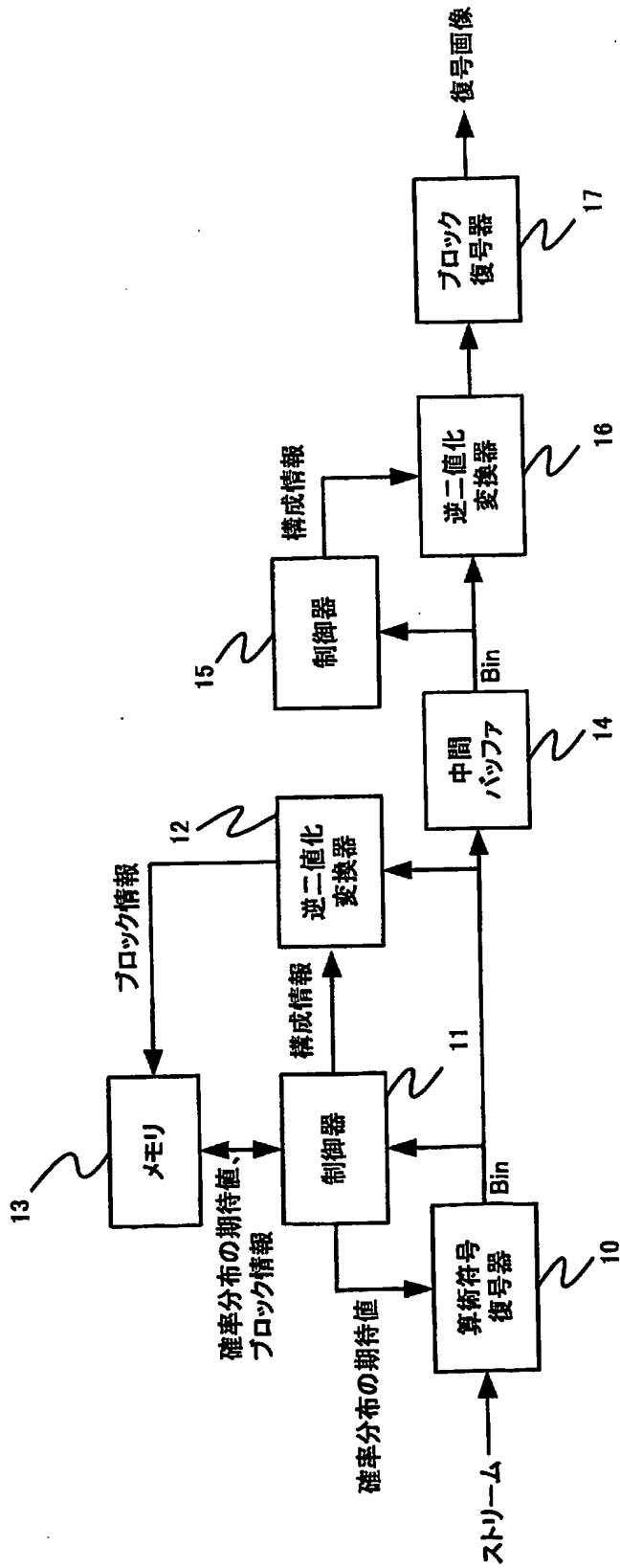
【0102】

- 10 算術符号復号器
- 11 制御器
- 12 逆二値化変換器
- 13 メモリ
- 14 中間バッファ
- 15 制御器
- 16 逆二値化変換器
- 17 ブロック復号器
- 20 ブロック符号化器
- 21 二値化変換器
- 22 中間バッファ
- 23 逆二値化変換器
- 24 制御器
- 25 算術符号化器
- 26 メモリ
- 27 ビット数推定器
- 31 処理部
- 32 処理部

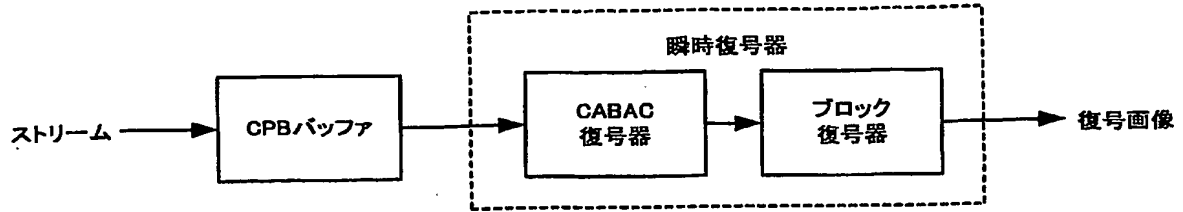
3 3 メモリ

【書類名】 図面

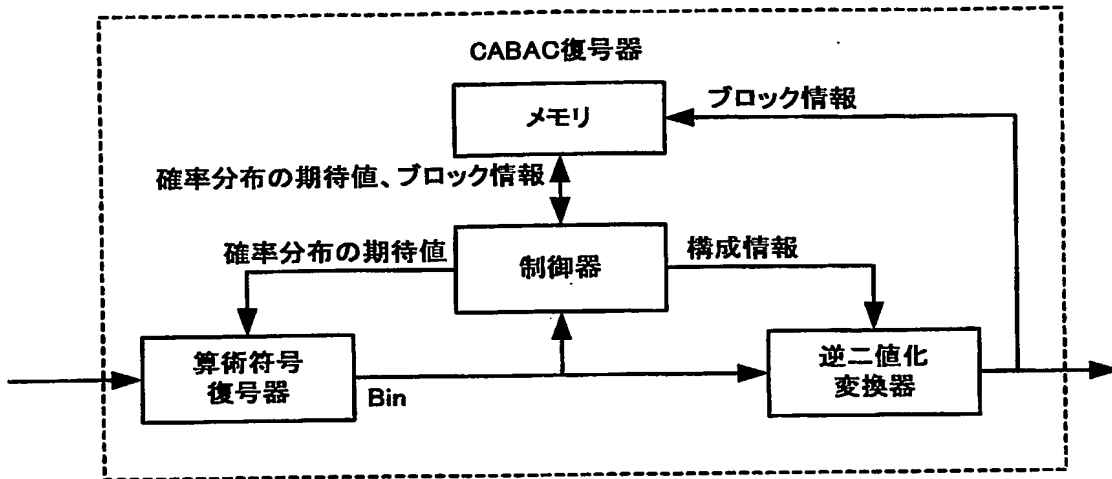
【図 1】



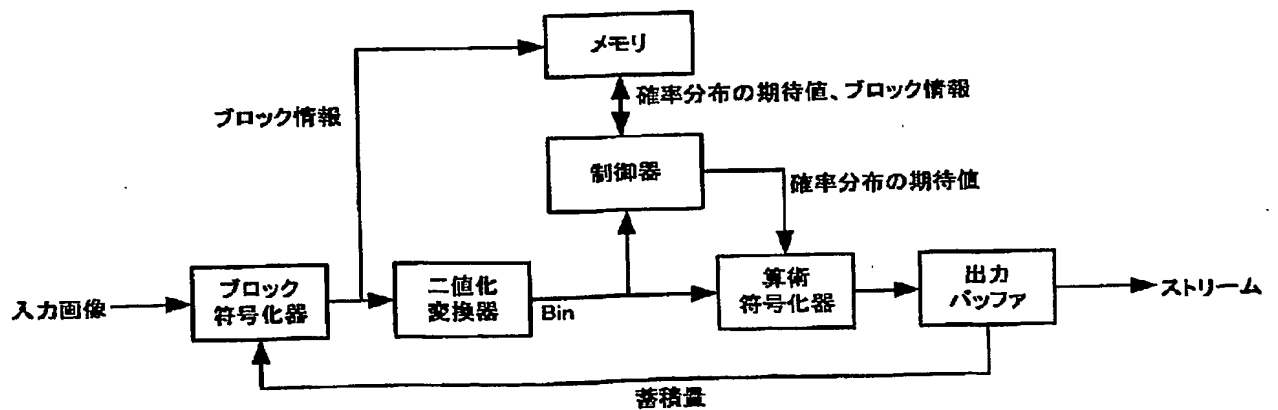
【図 2】



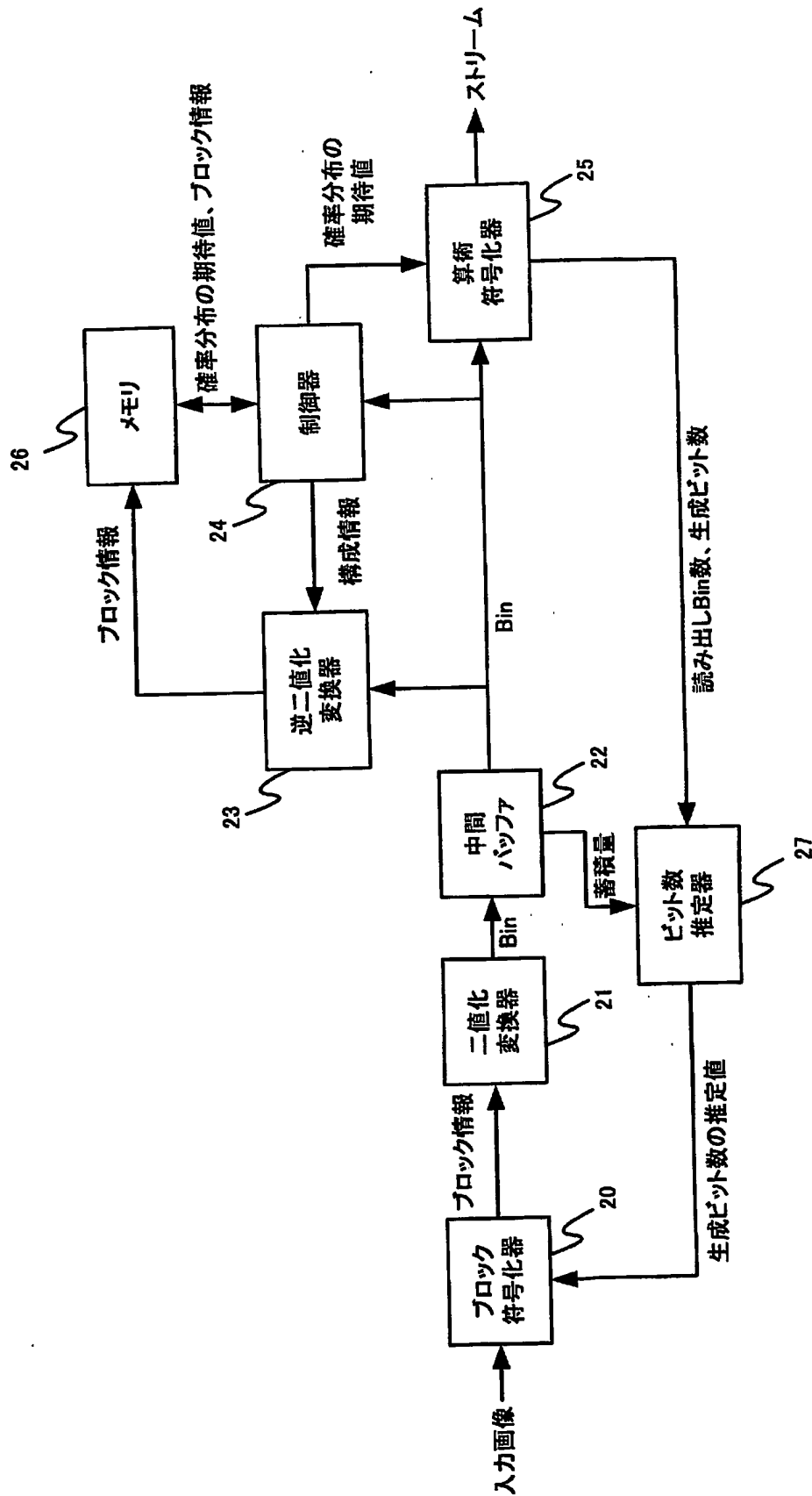
【図 3】



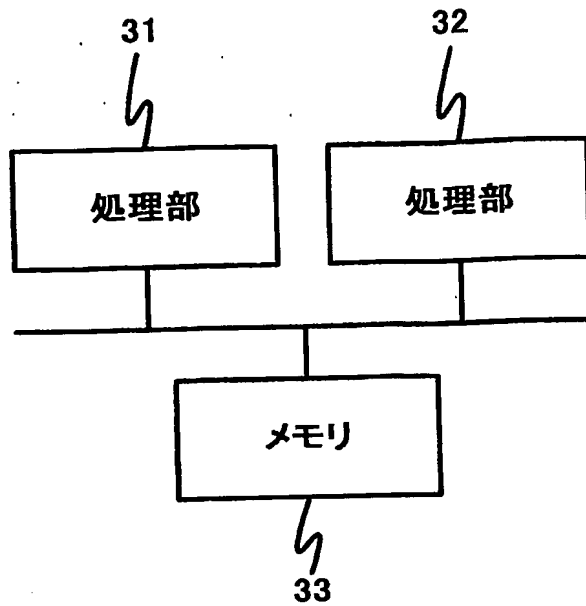
【図 4】



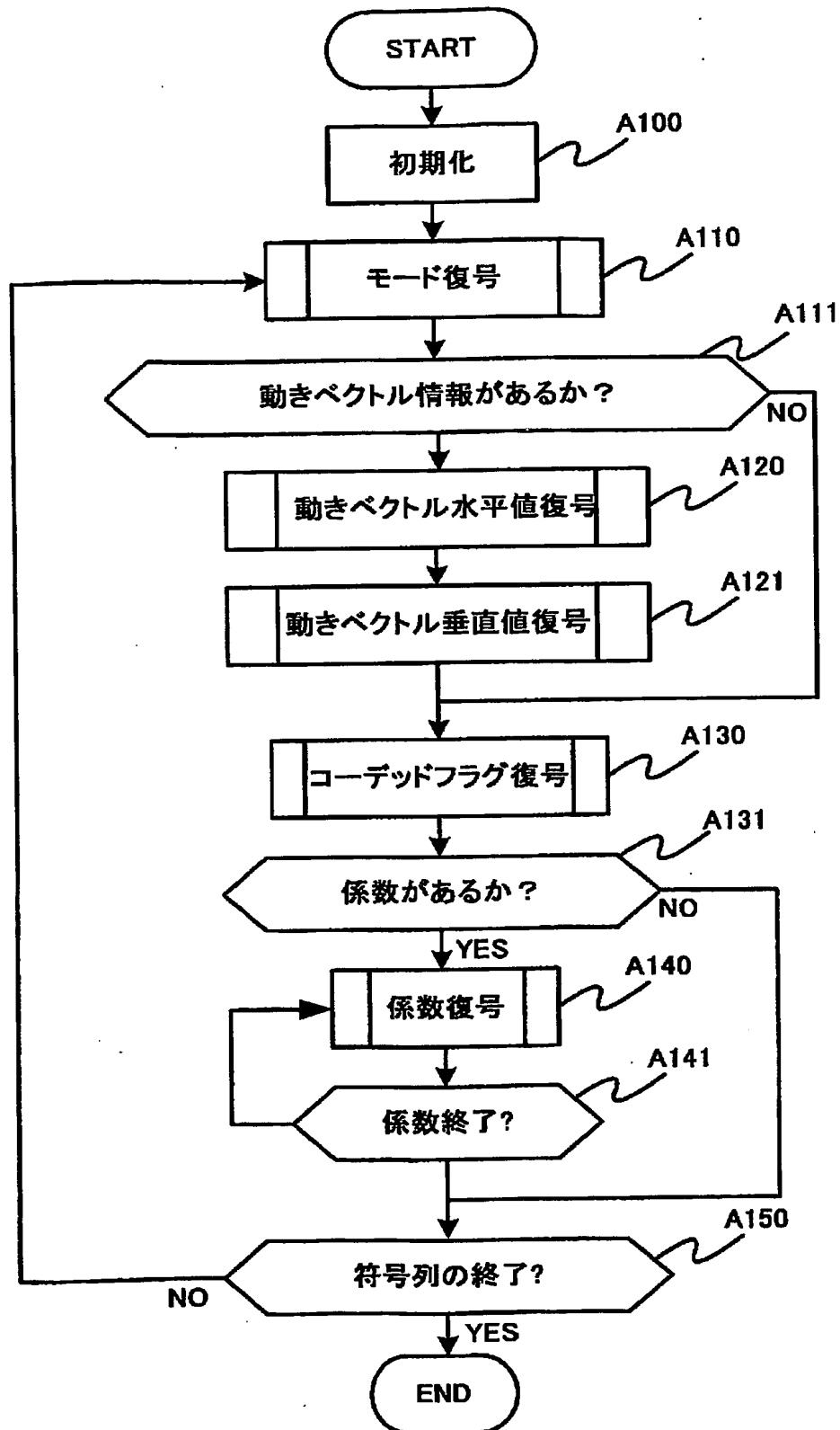
【図 5】



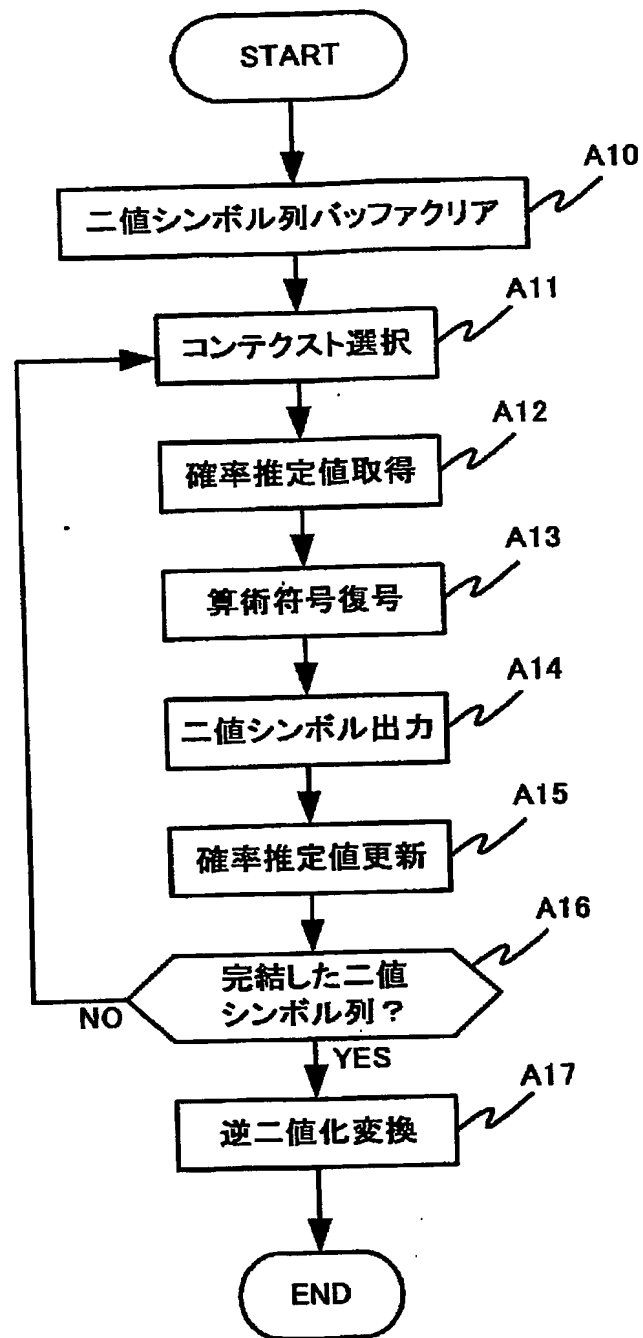
【図 6】



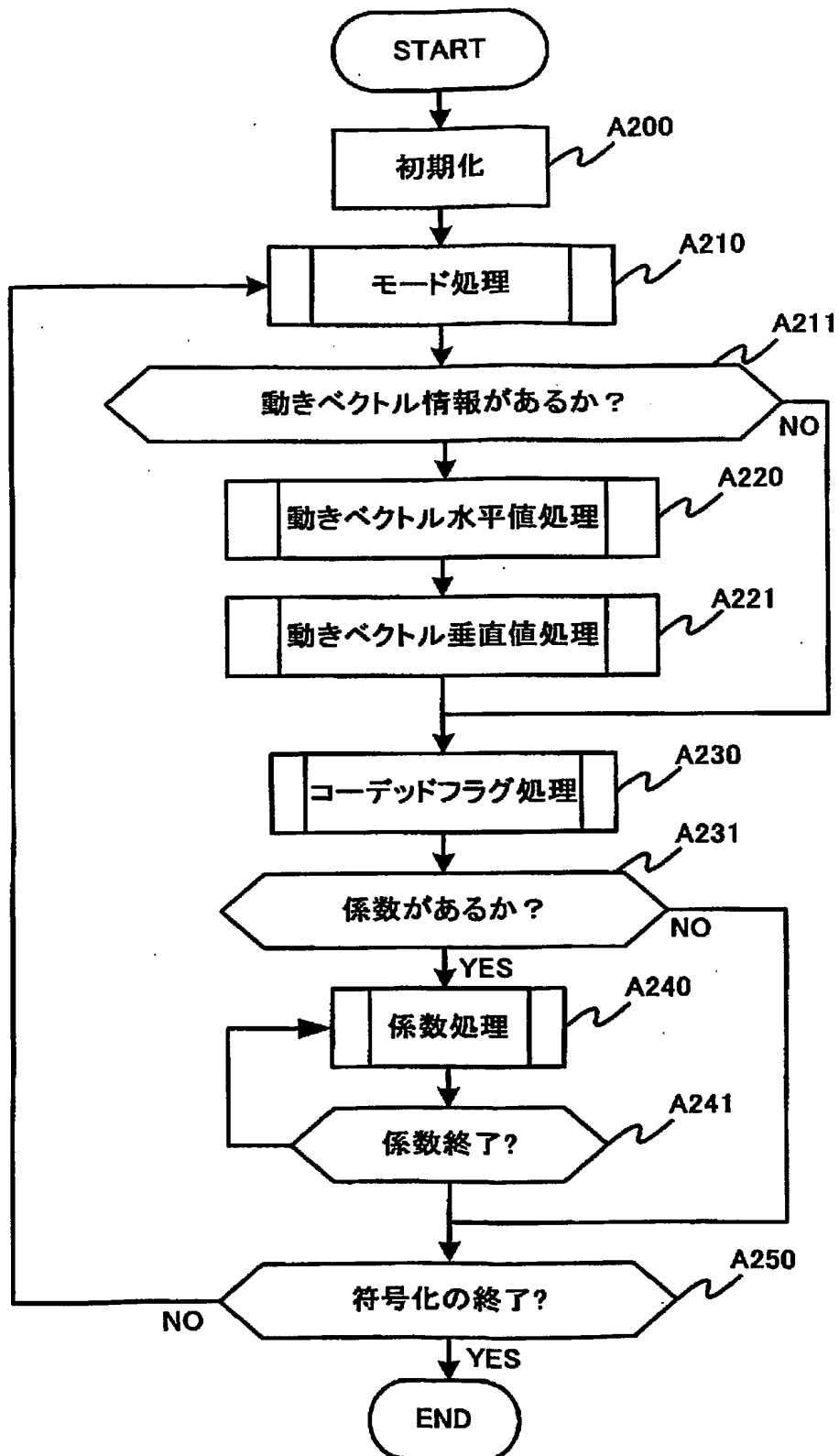
【図 7】



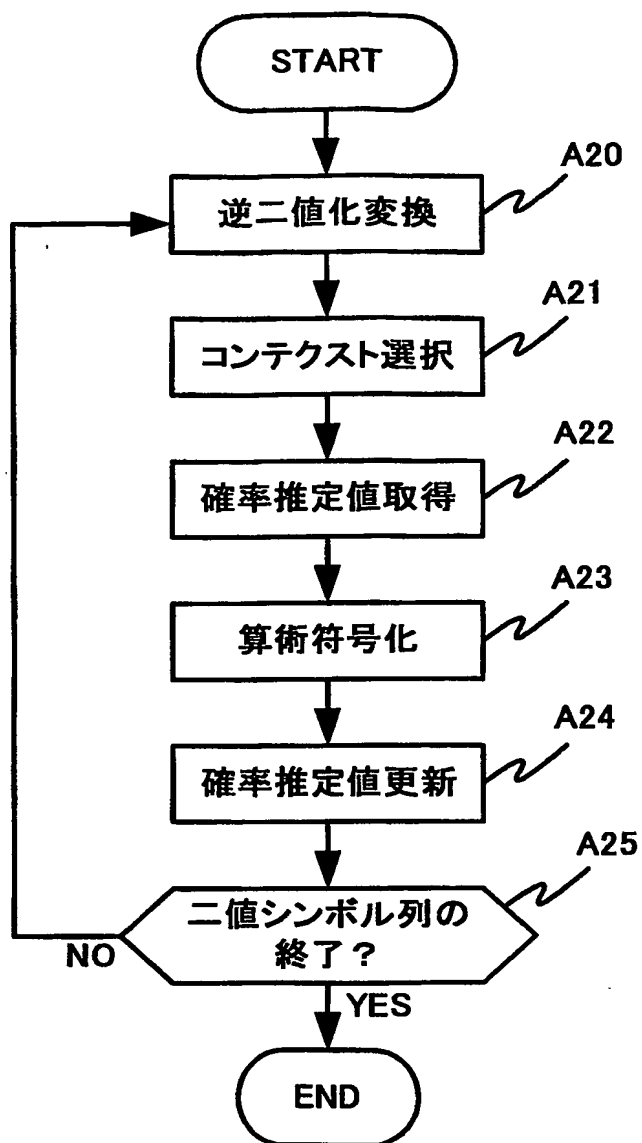
【図 8】



【図 9】



【図10】



【書類名】 要約書

【要約】

【課題】 二値化算術符号の復号器および符号化器の実装において、達成すべき二値シンボル処理レートの最大値が高いために、容易にもしくは安価に実現するのが困難であった。

【解決手段】 本発明の二値化算術符号の復号器では、算術符号の復号と逆二値化変換とを分離し、その間に大きな中間バッファを挿入する。まず、算術符号の復号はストリームが入力された時点で行う。これにより、算術符号は復号器の最大入力ビットレートで復号できればよくなる。得られた二値化シンボル列は一旦中間バッファに溜めておく。そして、二値化シンボル列から多値シンボルへの逆二値化変換は、後段のブロック復号器に処理に合わせて行う。

【選択図】 図 1

認定・付加情報

特許出願の番号	特願 2003-369176
受付番号	50301794773
書類名	特許願
担当官	第七担当上席 0096
作成日	平成15年10月30日

<認定情報・付加情報>

【提出日】 平成15年10月29日

特願 2 0 0 3 - 3 6 9 1 7 6

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 4 2 3 7]

1. 変更年月日

1 9 9 0 年 8 月 2 9 日

[変更理由]

新規登録

住 所

東京都港区芝五丁目 7 番 1 号

氏 名

日本電気株式会社